Using High Dimensional Indexes to Support Relevance Feedback Based Interactive Images Retrieval*

Junqi Zhang[1] Xiangdong Zhou[1†] Wei Wang[1] Baile Shi[1] Jian Pei[2]
[1] Fudan University, Shanghai, China
[2] Simon Fraser University, Burnaby, BC, Canada
{041021054, xdzhou, weiwang1, bshi}@fudan.edu.cn, jpei@cs.sfu.ca

ABSTRACT

Image retrieval has found more and more applications. Due to the well recognized *semantic gap* problem, the accuracy and the recall of image similarity search are often still low. As an effective method to improve the quality of image retrieval, the relevance feedback approach actively applies users' feedback to refine the search. As searching a large image database is often costly, to improve the efficiency, high dimensional indexes may help. However, many existing database indexes are not adaptive to updates of distance measures caused by users' feedback. In this paper, we propose a demo to illustrate the relevance feedback based interactive images retrieval procedure, and examine the effectiveness and the efficiency of various indexes. Particularly, audience can interactively investigate the effect of updated distance measures on the data space where the images are supposed to be indexed, and on the distributions of the similar images in the indexes. We also introduce our new B^+-tree-like index method based on cluster splitting and iDistance.

1. BACKGROUND

Image retrieval is important in many applications. Typically, in a similarity search, a user wants to search for images that are similar to a given query image. However, due to the well recognized *semantic gap* problem [1], the accuracy and the recall of image similarity search are often still low.

As an effective method to improve the quality of image retrieval, the *relevance feedback approach* [13] actively applies users' feedback to refine the search. In the first round, a (small) number of images are output for user examination. A user may give her/his feedback by confirming some output images are (or are not) similar to the query image. Then, the search system revises the search again. The feedback procedure can be conducted iteratively until the user is satisfied with the query results.

Searching a large image database is not cheap at all. In content-based image retrieval, images are often indexed by high dimensional feature vectors whose dimensionality varies from tens to thousands [8]. Thus, a similarity search can be computed by finding the K-nearest neighbors in the feature space. To speed up searches, a few multidimensional indexes were developed in previous studies, such as R-tree [5] and its variations and M-tree [4]. However, most of the existing index methods suffer from the curse of dimensionality, that is, their performance degrades on high dimensional data sets.

It is far from trivial to use indexes to support relevance feedback based interactive image retrieval. A major challenge is that, after the users' feedback is considered, the similarity (or distance) functions are often revised in the next round of search. Thus, an index structure based on a fixed similarity (or distance) measure has to be adaptive to the updated measure. If an index has to be updated largely for an updated measure, then the effectiveness of the index and the efficiency of the retrieval may suffer seriously.

To the best of our knowledge, QIC-M-tree [3] and VA-file [11] are the only two existing high dimensional index methods that can support relevance feedback image retrieval. Both methods do not update the indexes. Instead, they "map" the new similarity requirement using the updated measure back to the original space.

In the VA-file method [11], an upper bound of the similarity in the original distance measure using the updated distance function is computed after the feedback is taken in a round. Then, the upper bound is used as the filter to search the database. The search is implemented as one (sequential) scan of the VA-files. Only those images satisfying the bound are checked against the updated distance measure in the refinement step.

In the QIC-M-tree method [3], three types of distances are used. First, the index distance is used to construct an M-tree. Second, the query distance defined by users' queries is used to measure the similarity between the images indexed

*This work was partially supported by the NSF of China Grant 60403018, the NSF of Shanghai, China Grant 04ZR14011, 973-Plan Grant 2005CB321905, NSERC Discover Grant 312194-05, NSERC Grant 614067, and the NSF Grant IIS-0308001. All opinions, findings, conclusions and recommendations in this paper are those of the authors and do not necessarily reflect the views of the funding agencies.
†Corresponding author.

VLDB '06, September 12-15, 2006, Seoul, Korea.

and the query images. Last, the comparison distance is used to filter out branches that do not need to be searched. The comparison distance is a lower bound of the query distance. Thus, if an image (or a subset of images in a branch) has an estimated distance larger than the requirement, it can be pruned in the search since its query distance cannot satisfy the requirement. Moreover, a dimension reduction method is developed to compute the comparison distance efficiently.

The VA-file method assumes uniform data distribution. When the data set is not uniformly distributed, the efficiency of the VA-file is degraded. Many real data sets are not uniformly distributed. For instance, in the context of image retrieval, the distribution of image features is examined in [2].

Because of its strategy of splitting, the space efficiency of M-tree is low. For example, in a 32-dimensional real data set, searching using an M-tree is slower than using a sequential scan [7]. In QIC-M-tree, the filtering power of the lower bounding distance-based triangle inequality degrades rapidly. Although QIC-M-tree uses the comparison distance to conduct a "second" round filtering, the effectiveness of the comparison distance is not satisfactory due to the trade-off between the filtering capability and the computation cost [3]. Generally, the more dimensions are reduced, the weaker pruning power the comparison distance has.

Recently, we develop a new efficient method to use high dimensional index structures to support relevance feedback based interactive image retrieval. Based on the encouraging research and development results, in this paper we propose a demo to elaborate our interesting findings. In the rest of the paper, we shall highlight the major technical ideas and the strengths of our new method, and describe our demo plan.

2. HIGHLIGHTS OF OUR NEW METHOD

Indexing high dimensional data is an important and challenging problem that has been studied systematically and extensively. Typically, two types of methods exist [10]. The *space-partitioning methods*, such as grid-files, K-d B-trees, and quadtrees, index objects by partitioning the space recursively. The *data partitioning methods*, such as R-trees, X-trees, SR-trees, TV-trees, and hB-trees, partition objects into subsets recursively. One well-recognized challenge of high dimensional indexes is the *curse of dimensionality*: on data sets with a dimensionality of tens or higher, the query-answering performance of many index structures degrades dramatically, and can be even worse than a linear scan solution.

On data sets in metric spaces, the triangular inequality can help to speed up query answering. For example, in metric index tree structures such as M-trees, Slim-trees, VP-trees and GNAT, data is partitioned into excluding groups recursively. Then, the triangular inequality can be used to prune the unpromising objects in retrieval.

Recently, the iDistance measure is developed and a high dimensional index based on B^+-trees is proposed [12, 7]. The central idea is to cluster objects and find a reference point for each cluster. Then, the distance between an object

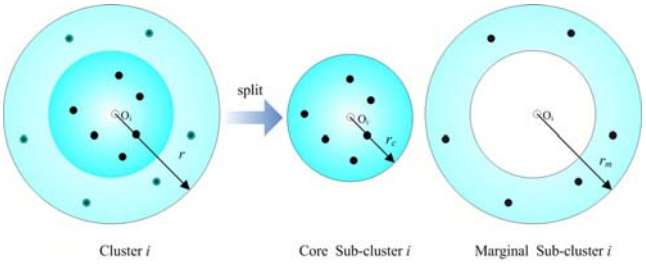

Figure 1: Cluster splitting.

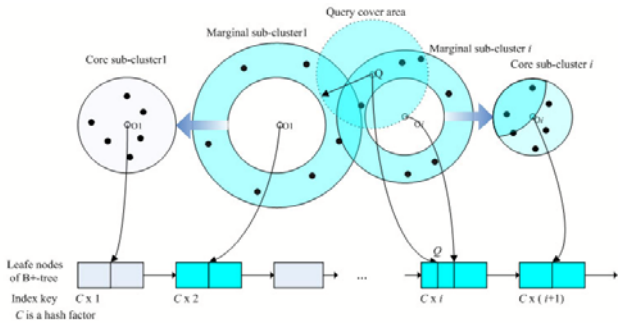

Figure 2: Cluster splitting based B^+-tree index structure

and the reference point in the cluster to which the object belong can be indexed in a B^+-tree. The iDistance search algorithm starts with a preset search radius and conducts nearest neighbor searches. It enlarges the search radius if necessary.

iDistance is an elegant method for high dimensional indexing and query answering. However, there are still some problems need to be solved if the method is to be applied for relevance feedback based image retrieval.

First, because the effectiveness of the high dimensional data clustering is still far from ideal, using clustering techniques directly to partition the data set may not lead to a good data filtering result. Second, in the context of relevance feedback based retrieval, we need to develop a good method to integrate users' feedback. Moreover, how to estimate the query radius is another important issue.

Our new approach borrows the idea of iDistance and the corresponding B^+-tree indexes. Technically, our approach has the following novel features.

First, we use K-means to cluster the data set. In real data sets, data in a cluster may not be uniformly distributed. Thus, it is not effective to search a whole cluster whenever the cluster intersects with the query region. Therefore, for each data cluster, we estimate the density of the data points, and select an appropriate ratio, such as 1:1 for data points in each part to split the cluster into two sub-clusters: the core and the marginal sub-cluster. Suppose m clusters are obtained from the K-means clustering. We have $2m$ clusters after the splitting. Figure 1 shows the idea for cluster splitting.

Figure 3: Demo system interface

Second, in the index construction, we use principle component analysis (PCA) to find the principle component with the maximum deviation in the data set, and use the best reference point in the component as the index reference. This is similar to the method in [9]. The best reference point in the principle component with the second largest deviation is used as the filtering reference to define the estimated distance. Then, the estimated distance is stored in the index, and does not need to be computed online.

Figure 2 illustrates the index structure. In the figure, cluster 1 is split into two clusters: the core and the marginal subclusters. They are stored on different segments of the leaf nodes according to the distances from the reference point. For the query Q in the figure, although the query covers an area intersecting with clusters 1 and 2, we do not need to search the core sub-cluster 1, since in fact the core subcluster 1 does not intersect with the query area. Thus, we can save some cost on similarity search.

Third, in interactive relevance feedback processing, the query distance is updated using users' feedback and the index distance is guaranteed to be a lower bound of the query distance. Thus, the index structure does not need to be changed.

Our extensive empirical study shows that our new approach can improve the efficiency of query answering substantially. Two real image data sets are used: data set $D1$, a collection of 32-dimensional color histogram feature vectors obtained from http://kdd.ics.uci.edu/databases/CorelFeatures/CorelFeatures.data.html. It has 68,040 images. Data set $D2$ contains 59,895 color images. For each image, a 16×3-dimensional RGB channel histogram and 10-dimensional wavelet coefficients are extracted to makeup the image feature vectors. The data set $D1$ is commonly used in many previous studies as a benchmark. The drawback of the data set $D1$ is that the images cannot be displayed, so we use it to examine the effectiveness of indexing, and use data set $D2$ to perform relevance feedback based interactive image retrieval experiments. Our test system is running on a PC with one P4 2.8GHz CPU and 512MB Memory.

K	10	20	30	40	50
M-tree	160	183	199	219	228
Linear scan	50	55	60	68	69
Our method w. 2-split	24	26	28	29	29
Our method w. 8-split	16	18	20	21	22

Figure 4: Query answering time without relevance feedback in milliseconds on data set $D1$.

Method	Search	Feedback
Linear scan	186	184
B+tree w.o. semantic bounding	166	172
Our semantic bounding method	166	94

Figure 5: Query answering time with relevance feedback in milliseconds on data set $D2$.

Figure 3 shows the interface. Each bar describes the relation of query image to one specific cluster. The cells in each bar denote clustering rings. The red cells contain relevant images, and the green ones do not contain relevant images. The red and green cells are accessed in the search process. The blue cells are not visited in the search process. Those cells are unpromising and filtered out by the index in the searching process. The feedback radius is the maximum distance from the query image to the images in the feedback set. The result radius is the maximum distance from the query image to the images in the result set. We get the feedback boundary by multiplying a constant factor S to the feedback radius, where S is a scalar used to guarantee the index distance being a lower bound of the query distance [6]. Our method uses this feedback boundary as the search radius in next round. However, in the traditional methods (e.g., [11]), the resulting boundary by multiplying S to the resulting radius should be used as the search radius.

The experiment results of K-NN search without relevance feedback are given in Figure 4. It shows the average query answering time of different methods based on $50,000$ random queries on the data set $D1$. According to [7], the iDistance method is about 2 times faster than a linear search. Figure 4 shows that our method with cluster 2-splitting is 2 times faster than a linear search. When each cluster is uniformly partitioned into 8 sub-clusters, our method is 3 times faster than a linear search.

The experimental results of relevance feedback based retrieval are given in Figure 5. It shows the average query answering time of different methods based on $1,000$ random queries on the data set $D2$. From the results, we can observe that, without a good strategy to make the index adaptive to updated distance measure, using B^+-tree does not help to improve the query answering efficiency. This is because the method of lower bounding distance used in the previous methods and the rigid B^+-tree may dramatically enlarge the search radius. Consequently, the refining process may be conducted on almost the whole data set, just like a linear scan search. We will show this phenomenon in our demo system. Our new approach uses a more restrict search radius to improve the search efficiency substantially. It is almost 2 times faster than the linear scan approach and the rigid B^+-tree method. The results strongly indicate the effectiveness of our method.

3. DEMO PLAN

We plan to present an image retrieval system in the demo. Particularly, we shall focus on the following aspects.

First, we shall bring to the demo some real data sets and demonstrate the relevance feedback based image retrieval procedure. The audience can understand the effectiveness of relevance feedback in the retrieval by playing with the system.

Second, we shall step by step elaborate our cluster splitting method, and show how it can improve the query efficiency. Audience are encouraged to use various feedback and inspect the corresponding changes on the query distance measure. The changes of the data spaces (i.e., the distribution of the images) will be visualized.

Third, to examine the role of indexes in the interactive relevance feedback based image retrieval, we shall bring to the demo our implementation of several methods, including a linear search method, the VA-file approach, and our new approach. When audience give feedback, how the indexes are affected will be computed and visualized. In particular, we shall illustrate and compare the distributions of the images similar to the query image in both the original distance and the updated distance. By such a comparison, audience can gain the insight of the challenges and the opportunities of indexes for interactive relevance feedback based image retrieval, and how the methods differ in performance.

4. REFERENCES

[1] A. Smeulders et al. Content-Based Image Retrieval at the end of the early years. In *IEEE Trans. On Pattern Analysis And Machine Intelligence*, Vol. 22, No. 12, December 2000.

[2] G. H. Cha and C. W. Chung. The GC-Tree: A High-Dimensional Index Structure for Similarity Search in Image Databases. *IEEE Transactions on Multimedia*, Vol. 4, No. 2, pp. 235-247,June 2002.

[3] P. Ciaccia And M. Patella Searching in Metric Spaces with User-Defined and Approximate Distances In*ACM Transactions on Database Systems* Vol. 27, No. 4, December 2002, Pages 398-437.

[4] P. Ciaccia et al. M-tree: An efficient access method for similarity search in metric spaces. In *VLDB'97*.

[5] A. Guttman. R-tree: A dynamic index structure for spatial searching. In *SIGMOD'84*.

[6] J. Hafner et al. Efficient color histogram indexing for quadratic form distance functions. In *IEEE Trans. Patt. Anal. Mach. Intell* ,17(7),pages 729-736. July 1995

[7] H. V. Jagadish et al. iDistance: An Adaptive B^+-tree Based Indexing Method for Nearest Neighbor Search. *ACM Transactions on Database Systems*, Vol. 30, No. 2, June 2005, Pages 364-397.

[8] W. -Y. Ma and B. S. Manjunath. Texture Features and Learning Similarity. In *CVPR'96*

[9] H. T. Shen, B.C. Ooi, X. Zhou. Towards Effective Indexing for Very Large Video Sequence Database. In *SIGMOD'05* .

[10] R. Weber et al. A Quantitative Analysis and Performance Study for Similarity-Search Methods in High-Dimensional Spaces In *VLDB'98*.

[11] P. Wu and B. S. Manjunath. Adaptive nearest neighbor search for relevance feedback in large image databases. In *Proc. 2001 ACM MM'01*.

[12] C. Yu et al. Indexing the distance: an efficient method to knn processing. In *VLDB'01*.

[13] X. S. Zhou and T. S. Huang. Relevance feedback for image retrieval: a comprehensive review. In *ACM Multimedia Systems Journal*, 8(6),pages 536–544, 2003.